**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Topic 42 - Functions: Passing Information**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**What**

In Python, a function can take **arguments** when it's called. Arguments are values or data given to the function so it can use them to perform tasks.

**Why**

Passing information into functions makes them more flexible and reusable. Instead of hard-coding specific values in the function, you can pass different data to achieve varied results with the same code.

**How**

1. **Basic Function with Arguments**:
   * Instead of hard-coding values inside the function, you can pass them as arguments when you call the function.

python

Copy code

def add\_numbers(first\_number, second\_number): # Function accepts two arguments

total = first\_number + second\_number

print(total)

add\_numbers(53, 109) # Call function and pass 53 and 109 as arguments

# Output: 162

1. **Parameters and Arguments**:
   * **Parameters** are variable names in the function definition (e.g., first\_number, second\_number).
   * **Arguments** are the values or variables you pass into the function call (e.g., 53, 109).
2. **Positional Arguments**:
   * Arguments are assigned to parameters based on their position in the function call.
   * The first argument goes to the first parameter, the second to the second, etc.

python

Copy code

add\_numbers(1.11, 2.22) # 1.11 -> first\_number, 2.22 -> second\_number

# Output: 3.33

1. **Variable Arguments**:
   * You can pass variables instead of hard-coded values.
   * The parameter catches the value of the variable passed.

python

Copy code

greeting = "Hello, there."

def greet\_user(message):

print(message)

greet\_user(greeting) # Pass greeting variable

# Output: Hello, there.

1. **Naming Parameters and Arguments**:
   * The names of arguments (in the call) and parameters (in the definition) don’t have to match.
   * However, it’s often helpful to give them matching names for clarity.

python

Copy code

whatever = "Hello, there."

def greet\_user(message):

print(message)

greet\_user(whatever) # Variable whatever is passed to parameter message

# Output: Hello, there.

**Things to Remember**

* You can pass different types of values, including numbers, strings, and variables, as arguments.
* Arguments are assigned to parameters in the order they’re given unless specified otherwise (as **positional arguments**).
* Keeping the names of arguments and parameters consistent can make your code easier to understand.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**